**auto、引用、常量**

1. 局部变量的初始化
   1. int a = 1;
   2. int a {1}; //c11引进的
2. auto（类型自动推断）
   1. 发生在编译期间，不会造成程序执行效率的降低
   2. 例如
      1. auto a = 1； //自动推断为 int
      2. auto a = ‘a’； //自动推断为 char
      3. 代码

#include <iostream>

#include <typeinfo>

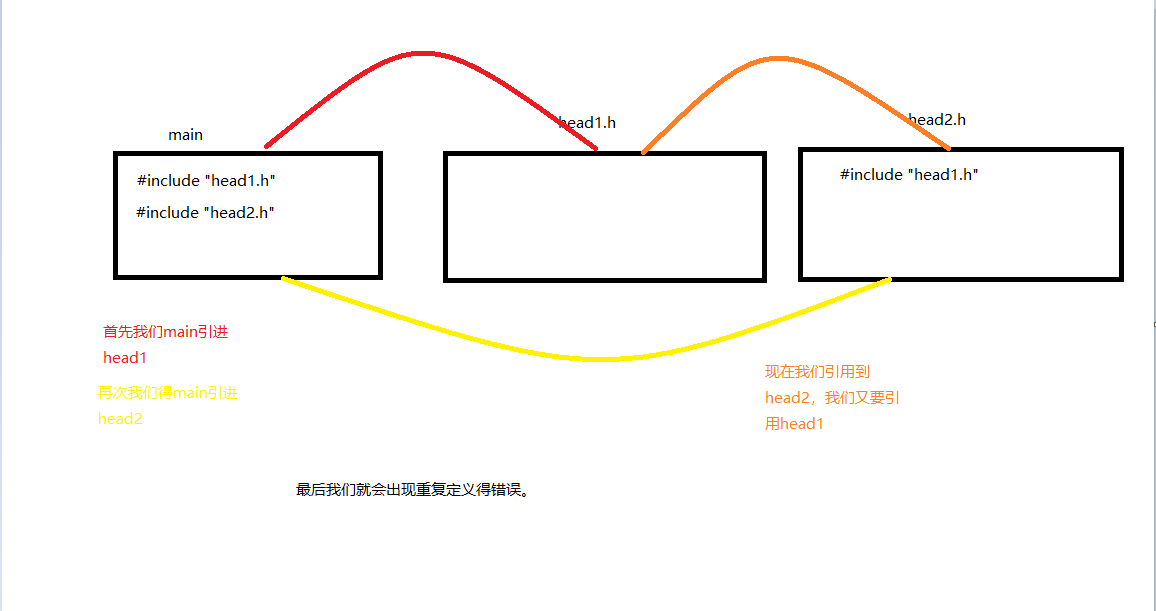
int main(){

auto i = 'a';

std::cout << typeid(i).name();

}

1. 头文件防卫
   1. 下面有一个例子



* 1. 那我们怎么解决这个问题呢
     1. 我们只需要在头文件前面加上 #pragma once
     2. 我们之前学过条件编译，我们也可以用条件编译来解决

#ifndef HEAD2\_\_H\_\_ //第一次从main函数调用时，未定义，所以定义执行下面得语句……等到下次调用到，我们之前调用过，所以就跳过，不执行

#define HEAD2\_\_H\_\_

#include "head1.h"

int g\_a = 10;

#endif // !HEAD2\_\_H\_\_

1. 引用（不能绑定到变量去，只能是变量）
   1. 相当于为变量取一个别名，用&符号表示，那么这别名和变量我们就可以看成是一个变量
   2. 代码

int a = 1;

Int &aa = a;

aa = 3； //所以 a 也变成3了

* 1. 2